

①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ Patentschrift
⑩ DE 41 34 207 C 1

⑤1 Int. Cl.⁵:
G 06 F 11/28
G 06 F 9/445
G 06 F 15/16

②1 Aktenzeichen: P 41 34 207.0-53
②2 Anmeldetag: 16. 10. 91
④3 Offenlegungstag: —
④5 Veröffentlichungstag
der Patenterteilung: 1. 4. 93

DE 41 34 207 C 1

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

⑦3 Patentinhaber:
ANT Nachrichtentechnik GmbH, 7150 Backnang, DE

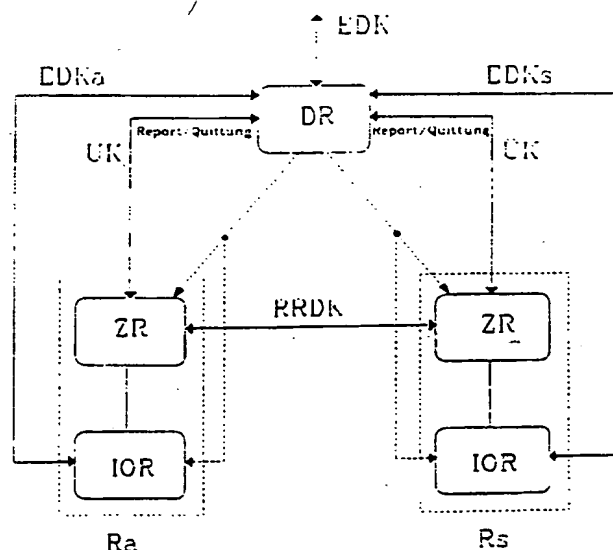
⑦2 Erfinder:
Harbich, Thomas, Dr.rer.nat., 7153 Weissach, DE;
Hirte, Jürgen, Dipl.-Ing., 7159 Auenwald, DE

⑤6 Für die Beurteilung der Patentfähigkeit
in Betracht gezogene Druckschriften:
US 43 71 754
Apel U., Online-Software-Erweiterung und
-Änderung, in: Elektrisches Nachrichtenwesen
Bd. 64, No. 4, 1990, S. 327-333;

⑤4 Verfahren zum Laden eines Doppelrechner-Standby Systems

⑤7 Verfahren zum Laden eines Doppelrechner-Standby-Systems mit neuer Anwendersoftware, wobei die beiden Rechner über einen internen Datenkanal miteinander verbunden und jeweils über einen externen Datenkanal mit einer externen Datenquelle verbindbar sind und jeweils über ein Multitaskingbetriebssystem verfügen, mit einer Überwachungseinheit, durch welche die beiden Rechner wechselseitig in einen aktiven bzw. passiven Zustand geschaltet werden können, gekennzeichnet durch die Schritte Vorbereitung des passiven Rechners auf das Laden, Laden von einer externen Datenquelle aus über den externen Datenkanal, in den aktiven Rechner und von dort über den internen Datenkanal in den passiven Rechner, nach Ende des Ladens Starten des passiven Rechners und Umschaltung desselben in den aktiven Zustand, wenn neue Anwendersoftware fairlos.

Anschließend Laden des jetzigen passiven Rechners von dem aktiven Rechner über den internen Datenkanal. Keine Unterbrechung oder Beeinträchtigung des laufenden Betriebs durch das Laden. Rücklademöglichkeit der alten Anwendersoftware, wenn neue Anwendersoftware fehlerhaft.



BEST AVAILABLE COPY

Beschreibung

Die Erfindung bezieht sich auf ein Verfahren zum Laden eines Doppelrechner Standby-Systems gemäß Oberbegriff des Anspruches 1.

In dem US Patent 43 71 754 wird ein automatisches fehlerbehebendes Rechnersystem beschrieben, in dem aktive Speicherbereiche an Standby-Speichern geladen werden und zwischen aktiven und Standby-Kopien umgeschaltet wird. Dieses dient zur Fehlererkennung und Fehlerbehebung im laufenden Betrieb.

Der Aufsatz "Online-Software-Erweiterung und -Änderung" von U. Apel in Elektrisches Nachrichtenwesen Band 64 Nr. 4 von 1990 (Seite 327 — 333) beschäftigt sich hingegen mit dem Laden eines Rechners mit geänderter oder erweiterter Software, wobei die beiden Fälle unterschieden werden, ob die alte und die neue Software parallel laufen können oder nicht. Ist ersteres der Fall, so können die neue Komponente gestartet und Nachrichten an beide Komponenten gerichtet werden, wobei der Verkehr zu der neuen Komponente allmählich erhöht werden kann, bis die alte Komponente keine neuen Nachrichten mehr erhält und entfernt werden kann. Dieses Verfahren erfordert einen sehr großen Speicherbedarf.

Wenn die alte und die neue Komponente nicht parallel laufen können, so werden die alte Komponente auf die Funktionserweiterung vorbereitet und der erforderliche Zustandstransfer durchgeführt, wobei beim Erreichen vordefinierter Synchronisationspunkte angehalten oder geendet wird und alle Nachrichten zur neuen Komponente umgeleitet werden, die dann gestartet wird, wonach die alte Komponente schließlich beendet und entfernt wird. Bei diesem Verfahren übernimmt die neue Komponente sofort den gesamten Verkehr.

Nachteilig ist hierbei die Unterbrechung des laufenden Betriebs beim Laden der neuen Anwendersoftware.

Bei beiden Verfahren ist außerdem nachteilig, daß ein fehlerhafter Ladevorgang oder das Laden einer nicht ablauffähigen Anwendersoftware zum Systemausfall führt.

Der Erfindung lag die Aufgabe zugrunde, ein Verfahren der eingangs genannten Art anzugeben, welches in der Lage ist, eine Änderung oder Erweiterung der Anwendersoftware ohne Unterbrechung des laufenden Betriebes in unaufwendiger Weise durchzuführen. Außerdem soll ein fehlerhafter Ladevorgang oder das Laden einer nichtablauffähigen neuen Anwendersoftware nicht zum Systemausfall führen.

Diese Aufgabe wurde gelöst durch die Merkmale des Anspruches 1. Vorteilhafte Ausgestaltungen ergeben sich durch die Unteransprüche. Das erfindungsgemäße Verfahren weist die Vorteile auf, daß das Laden neuer oder erweiterter Anwendersoftware ohne Unterbrechung oder Beeinträchtigung des laufenden Betriebes ermöglicht wird. Der Ladevorgang erfolgt über den externen Datenkanal im fehlerfreien Übertragungsfalle nur ein einziges Mal, so daß die externe Datenquelle, welche die neue Anwendersoftware liefert, und der externe Datenkanal nur minimal belastet sind. Diese Belastung wird noch weiter reduziert, wenn lediglich die Änderung der neuen Anwendersoftware übertragen wird.

Wenn zum Laden des passiven Rechners die Zwischenspeicherung der zu übertragenden Anwendersoftware im Arbeitsspeicher des aktiven Rechners blockweise oder in einigen Blöcken erfolgt, so ist auch die Anforderung an die Speicherkapazität der beiden Rech-

ner nur minimal oder unwesentlich größer.

Nach Beendigung des Ladens des passiven Rechners wird die neue Software gestartet, und der Diagnose-rechner oder eine sonstige Überwachungseinheit kann dann feststellen, ob die neue Anwendersoftware fehlerfrei ist oder nicht. Sofern ein Fehler vorliegt, kann entschieden werden, ob der Ladevorgang wiederholt wird oder ob die alte Anwendersoftware wieder in den passiven Rechner geladen wird. Transiente Softwarefehler können repariert werden, indem die Anwendersoftware des einen Rechners automatisch durch diejenige des anderen Rechners geladen wird.

Es folgt nun die Beschreibung der Erfindung anhand der Figuren.

Die Fig. 1 zeigt ein grobes Blockschaltbild des Doppelrechnersystems.

In Fig. 2 sind die aktiven Programme und ihre Kommunikationspfade schematisch dargestellt.

Fig. 3 schließlich zeigt in Form eines Flußdiagramms den logischen Ablauf des erfindungsgemäßen Verfahrens.

Die Fig. 4 zeigt ein detaillierteres Blockschaltbild für ein Ausführungsbeispiel eines Doppelrechnersystems.

Die schematische Darstellung der Fig. 1 zeigt ein Doppelrechnersystem mit zwei identischen Rechnern 1 und 2, die über einen internen Datenkanal RRDK miteinander verbunden sind. Zusätzlich sind beide Rechner über je einen externen Datenkanal EDK mit einer übergeordneten Kontrollinstanz bzw. externen Datenquelle verbunden bzw. verbindbar; d. h. nur einer der beiden externen Datenkanäle ist für die Datenübertragung mit oder von der Kontrollinstanz offen, und zwar ist dies derjenige externe Datenkanal welcher zu dem Rechner führt, der im Augenblick für alle durchzuführenden Aufgaben zuständig ist. Dieser Rechner wird als aktiver Rechner bezeichnet, während der andere Rechner, welcher momentan keine Aufgaben bearbeitet, als Standby-Rechner bezeichnet wird. Die Entscheidung darüber, welcher der Rechner aktiv bzw. standby geschaltet ist, wird von einer übergeordneten Kontrollinstanz, von einem Diagnoserechner oder einer ähnlichen Überwachungseinheit getroffen. Die Umschaltung erfolgt durch entsprechende Kommandos, welche an beide Rechner gesendet werden. Bei einer Umschaltung verändert sich auch der Zustand der externen Datenkanäle EDK.

In Fig. 2 ist eine schematische Darstellung der auf den Rechnern aktiven Programme und ihre Kommunikationspfade gezeichnet. Beide Rechner sind mit einem Multitaskingbetriebssystem ausgestattet. Programme, die auf den Rechnern ausgeführt werden, werden als Prozesse gezeichnet. Jedem Prozeß wird eine individuelle Ablaufpriorität zugeordnet, wodurch bestimmte Prozesse gegenüber anderen bevorzugt bearbeitet werden. Diejenigen Prozesse, welche für das Laden einer neuen Softwareversion zuständig sind, werden unter dem Begriff Download-Software DLSW zusammengefaßt. Sie wird mit einer Ablaufpriorität ausgestattet, die garantiert, daß andere Prozesse, welche betriebliche Aufgaben bearbeiten, also die sogenannten Applikationen Appl, bevorzugt werden. Dadurch wird garantiert, daß der laufende Betrieb der Applikationen, nämlich Führung und Steuerung der Peripherie, nicht durch das Laden einer neuen Softwareversion beeinträchtigt oder gar unterbrochen wird. Über den externen Datenkanal EDK können von einer übergeordneten Kontrollinstanz Kommandos an den aktiven Rechner gesendet werden. Mittels eines Parsers wird entschieden bzw. festgestellt, ob diese Kommandos für die Applikationen

Appl oder für die Download-Software DLSW bestimmt sind. Der Entscheidungsprozeß erfolgt dabei so, daß die Adresse der empfangenen Kommandos ausgewertet und die Kommandos entsprechend an den jeweiligen Adressaten weitergeleitet werden. Die Download-Software DLSW auf dem aktiven Rechner RA hat die Aufgabe, die von der Kontrollinstanz empfangenen Kommandos über den internen Rechnerrechnerdatenkanal RRDK an die Download-Software DLSW des Standby-Rechners weiterzusenden und durch kontinuierliches Austauschen von Kommandos mit der Kontrollinstanz die Funktionstüchtigkeit des aktiven Rechners und seiner Software anzuzeigen. Die Download-Software DLSW des Standby-Rechners hat die Aufgabe, seine Programme auf das Laden einer neuen Softwareversion vorzubereiten, diese zu laden und zu starten. Sowohl für das Veranlassen der Vorbereitung auf das Laden einer neuen Softwareversion als auch zur Durchführung desselben werden Kommandos von der Kontrollinstanz über die Download-Software DLSW des aktiven Rechners an die Download-Software DLSW des Standby-Rechners übertragen.

Die Fig. 3 zeigt in Form eines Flußdiagramms den logischen Ablauf des erfindungsgemäßen Verfahrens zum Einspielen einer neuen Softwareversion auf ein Doppelrechnersystem. Im Ausgangszustand 1 befinden sich beide Rechner in einem Betriebszustand, in dem mit der alten Softwareversion gearbeitet wird. Als erster Schritt des Ladevorgangs wird ein Kommando von der Kontrollinstanz oder übergeordneten Instanz über den offenen externen Datenkanal EDK an die Download-Software DLSW des aktiven Rechners gesendet; letztere leitet das Kommando über den internen Datenkanal RRDK an die Download-Software DLSW des Standby-Rechners weiter. Die Prozesse auf dem Standby-Rechner werden daraufhin auf das bevorstehende Laden einer neuen Anwendersoftwareversion vorbereitet. 2. Das Laden 3 der neuen Softwareversion erfolgt von einer externen Datenquelle, die sich auch bei der übergeordneten Kontrollinstanz befinden kann, über den aktiven Rechner auf den Standby-Rechner. Dazu werden die zu übertragenden Daten in Form von Kontrollkommandos, welche die Daten enthalten, an die Download-Software DLSW auf den aktiven Rechner gesendet. Ist die neue Softwareversion auf dem Standby-Rechner vollständig geladen, so gibt es zwei Möglichkeiten. Erstens von der Kontrollinstanz wird ein Umschalten 4 (gemäß Fig. 3) der Rechnerzustände vorgenommen. Der zuvor aktive Rechner mit der alten Softwareversion geht in den Zustand standby über, und der im Zustand standby befindliche Rechner mit der neuen Softwareversion geht entsprechend in den Zustand aktiv über. Stellt sich nun heraus, daß die neue Softwareversion fehlerhaft 9 ist, so wird ein sofortiges Rückschalten der Rechnerzustände 10 bewirkt. Die Rückschaltung muß natürlich so schnell erfolgen, daß eine Beeinträchtigung der Betriebsabläufe nicht erfolgt.

Die zweite Möglichkeit (nicht gezeichnet in Fig. 3) sieht vor, daß vor einer Umschaltung der Rechnerzustände die neue Softwareversion auf dem Standby-Rechner gestartet und beispielsweise mittels einer Diagnose oder Überwachung auf Fehler untersucht wird. Bei Fehlerfreiheit kann dann eine Umschaltung so erfolgen, daß die Bearbeitung von Aufgaben auf dem aktiven Rechner kontrolliert eingestellt wird. Wenn alle in Bearbeitung befindlichen Aufgaben der Applikationen erledigt sind, wird die Verbindung zu der Kontrollinstanz über den externen Datenkanal EDK abgebaut und um-

geschaltet. Der nach der Umschaltung aktive Rechner baut die Verbindung zur Kontrollinstanz wieder auf und signalisiert damit, daß die neue Softwareversion korrekt gestartet wurde. 5. Nunmehr wird der jetzige Standby-Rechner, beispielsweise durch ein Kommando der Kontrollinstanz, auf das bevorstehende Laden der neuen Softwareversion vorbereitet. 6. Das Laden derselben erfolgt, indem sie direkt vom aktiven Rechner über den internen Datenkanal RRDK in den Standby-Rechner übertragen wird. 7. Diese Verfahrensweise entlastet generell den externen Datenkanal EDK und ist natürlich von Vorteil, wenn die Übertragungskapazität des externen Datenkanals gegenüber derjenigen des internen Datenkanals unterlegen ist. Nach Beendigung des Ladevorganges sind beide Rechner mit einer neuen Softwareversion ausgestattet und befinden sich im Betrieb 8.

Über den Fall, daß die neue Softwareversion per se fehlerhaft ist, wobei z. B. der Ladevorgang korrekt und fehlerfrei erfolgt ist, so kann beispielsweise durch die übergeordnete Kontrollinstanz entschieden werden, daß der Standby-Rechner wieder mit der alten Softwareversion geladen wird. Er wird darauf vorbereitet, 11. worauf im Vorgang 12 diese über den internen Datenkanal RRDK vom aktiven auf den Standby-Rechner übertragen wird. Erfolgt dies fehlerfrei, so befinden sich beide Rechner wieder im Betriebszustand 13 mit der alten Softwareversion.

Beim Laden einer neuen Softwareversion können grundsätzlich zwei Vorgehensweisen unterschieden werden. Die eine geht davon aus, daß die neue Softwareversion komplett von der externen Datenquelle bzw. Kontrollinstanz auf den Standby-Rechner übertragen wird. Dies ist dann notwendig, wenn es sich um eine neue Softwareversion handelt, die keine alten Teile oder Moduln wiederverwenden kann.

Die zweite Vorgehensweise geht davon aus, daß bei einer neuen Softwareversion lediglich ein kleiner Teil geändert wurde, der Rest jedoch unverändert bleibt. In diesem Fall ist es möglich, lediglich den gegenüber der alten Version geänderten Teil bzw. das geänderte Modul der neuen Softwareversion von der Kontrollinstanz bzw. externen Datenquelle auf den Standby-Rechner zu übertragen. Die restlichen unveränderten Teile der alten Softwareversion werden dann vom aktiven Rechner über den internen Datenkanal auf den Standby-Rechner übertragen. Bei einer solchen Verfahrensweise wird der externe Datenkanal EDK durch den Ladevorgang weniger belastet.

Die bisherige Betrachtungsweise ging davon aus, daß der mit neuer Anwendersoftware zu ladende Standby-Rechner in einen Zustand zurückgesetzt wird, indem nur noch das Betriebssystem bzw. die für den Ladevorgang notwendigen Programmmoduln nicht gelöscht sind. Es sind natürlich auch Verfahrensweisen denkbar, bei denen eine totale Zurücksetzung nicht erfolgt, sondern bei denen lediglich Teile oder Moduln der alten Anwendersoftware ersetzt werden durch eine neue geänderte Version. Beim Ladevorgang muß dann natürlich eventuell die unterschiedliche Länge der Softwareteile der alten und neuen Version berücksichtigt werden, was zu einer Verschiebung im Speicher oder verteilten Einspeicherung führen kann. Eine solche Vorgehensweise entlastet den Verkehr sowohl auf dem externen als auch auf dem internen Datenkanal.

Die Fig. 4 zeigt ein detaillierteres Blockschaltbild für ein Ausführungsbeispiel einer Doppelrechneranordnung. Das Doppelrechnersystem besteht aus dem linken aktiven Rechner RA und dem rechten passiven Rechner

Rs. Beide Rechner sind wiederum Doppelrechnersysteme in der Architektur eines Hauptrechners oder zentralen Rechners ZR, dem ein Input-Outputrechner IOR eigens für Ein-/Ausgabe-Aufgaben vorgeschaltet ist. Beide Hauptrechner sind über einen Überwachungskanal ÜK mit einer Überwachungseinheit oder hier einem Diagnoserechner DR verbunden. Über diesen Überwachungskanal werden Report- bzw. Quittungssignale ausgetauscht. Die Überwachung schließt natürlich jeweils den Ein-Ausgaberechner IOR mit ein. Die Kontrollinstanz ist hier der Diagnoserechner DR, durch den auch die Umschaltung der Rechner in den aktiven bzw. passiven Zustand erfolgt, angedeutet durch die gestrichelten Linien, welche sowohl zum zentralen Rechner ZR als auch zum vorgeschalteten Ein-Ausgaberechner IOR führen.

Die Datenkanäle EDKa und EDKs führen auch jeweils über den Ein-/Ausgaberechner IOR. In der dargestellten Realisierung verbindet der Diagnoserechner DR den Datenkanal EDKa des augenblicklich aktiven Zentralrechners (ZR_a) mit dem externen Datenkanal EDK. Auf diese Weise muß die übergeordnete Kontrollinstanz lediglich eine Schnittstelle bedienen und das gesamte in Fig. 4 dargestellte Rechnersystem stellt sich der übergeordneten Kontrollinstanz wie ein Einprozessorsystem dar.

Patentansprüche

1. Verfahren zum Laden eines Doppelrechner-Standby Systems mit neuer Anwendersoftware, wobei die beiden Rechner über einen internen Datenkanal miteinander verbunden und jeweils über einen externen Datenkanal mit einer externen Datenquelle verbindbar sind und jeweils über ein Multitaskingbetriebssystem verfügen, mit einer Überwachungs- oder Diagnoseeinheit, durch welche die beiden Rechner wechselseitig in einen aktiven bzw. einen passiven Zustand schaltbar sind, gekennzeichnet durch die folgenden Schritte:

- a) der passive Rechner wird auf das Laden vorbereitet,
- b) die neue Anwendersoftware wird von der externen Datenquelle über den externen Datenkanal in den Speicher des aktiven Rechners übertragen,
- c) die neue Anwendersoftware wird vom aktiven Rechner über den internen Datenkanal in den Speicher des passiven Rechners übertragen,
- d) nach Beendigung dieser Übertragung wird die in Schritt c) geladene Anwendersoftware des passiven Rechners gestartet,
- e1) läuft die neue Anwendersoftware fehlerlos, so wird der bis dahin passive Rechner in den aktiven Zustand umgeschaltet und der bis dahin aktive Rechner in den passiven Zustand umgeschaltet, und es folgt Schritt f),
- e2) im Fehlerfall unterbleibt die Umschaltung, und es folgt Schritt h),
- f) es folgen die Schritte a), c) und d),
- g1) läuft die neue Anwendersoftware fehlerfrei, ist der Ladevorgang beendet,
- g2) im Fehlerfall werden die Schritte f) und g) wiederholt;
- h) durch eine übergeordnete Kontrollinstanz wird entschieden, ob die Schritte a) und e) wiederholt werden oder ob die alte Anwender-

software vom aktiven Rechner wieder in den passiven Rechner geladen wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Vorbereitung auf das Laden des passiven Rechners durch das Ladeprogramm des aktiven Rechners erfolgt, indem eine Kommunikation zwischen ihren beiden Ladeprogrammen aufgebaut wird.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß das Ladeprogramm des aktiven Rechners per Befehl über den externen Datenkanal zur Vorbereitung auf das Laden des passiven Rechners (Schritt a)) veranlaßt wird.

4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Übertragung der neuen Anwendersoftware gemäß Schritt b) durch blockweise Zwischenspeicherung der zu übertragenden Anwendersoftware im Arbeitsspeicher des aktiven Rechners erfolgt, daß anschließend der zwischengespeicherte Block gemäß Schritt c) in den passiven Rechner geladen wird und daß die Schritte b) und c) gemäß der Anzahl der zu übertragenden Blöcke wiederholt werden.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Laden während des laufenden Betriebes erfolgt.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß das Laden mit einer niedrigen Priorität bearbeitet wird.

7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß der Ladevorgang in den passiven Rechner gemäß Verfahrensschritt c) durch das Ladeprogramm des aktiven Rechners erfolgt, indem eine Kommunikation zwischen ihren beiden Ladeprogrammen aufgebaut wird.

8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die übergeordnete Kontrollinstanz die Überwachungs- oder Diagnoseeinheit ist.

9. Verfahren nach einem der vorhergehenden Ansprüche 1 bis 7, dadurch gekennzeichnet, daß die übergeordnete Kontrollinstanz die externe Datenquelle ist.

10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß, wenn die neue Anwendersoftware lediglich einen Teil oder einen Modul der gesamten Anwendersoftware betrifft oder umfaßt, nur dieser Teil bzw. Modul über den externen Datenkanal zum aktiven Rechner übertragen wird.

11. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß der passive Rechner mittels des Ladeprogramms des aktiven Rechners mit neuen und/oder alten Anwendersoftwaremodulen ladbar ist.

12. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß der passive Rechner zur Vorbereitung auf das Laden mit Ausnahme seines Betriebssystems in einen gelöschten Zustand zurückgesetzt wird.

13. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß am Anschluß des externen Datenkanals vor dem Rechner ein Input-Output-Rechner (IOR) eingefügt ist.

Hierzu 4 Seite(n) Zeichnungen

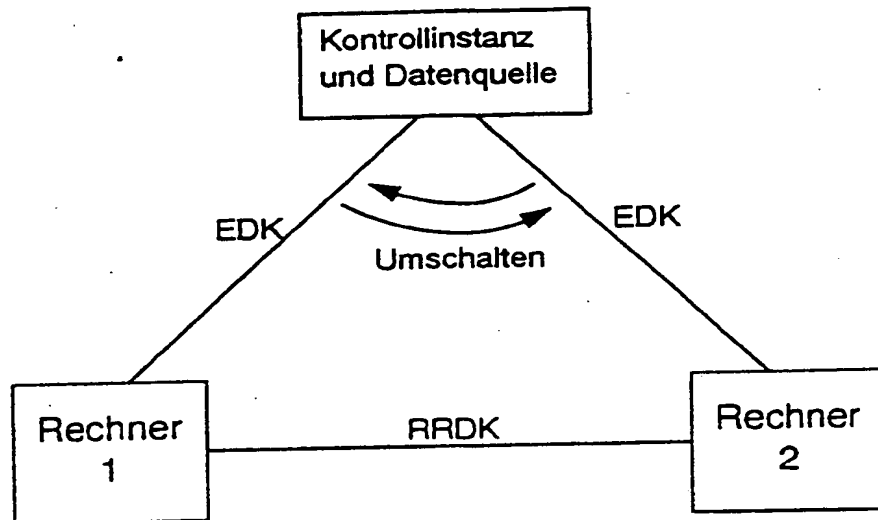


FIG. 1

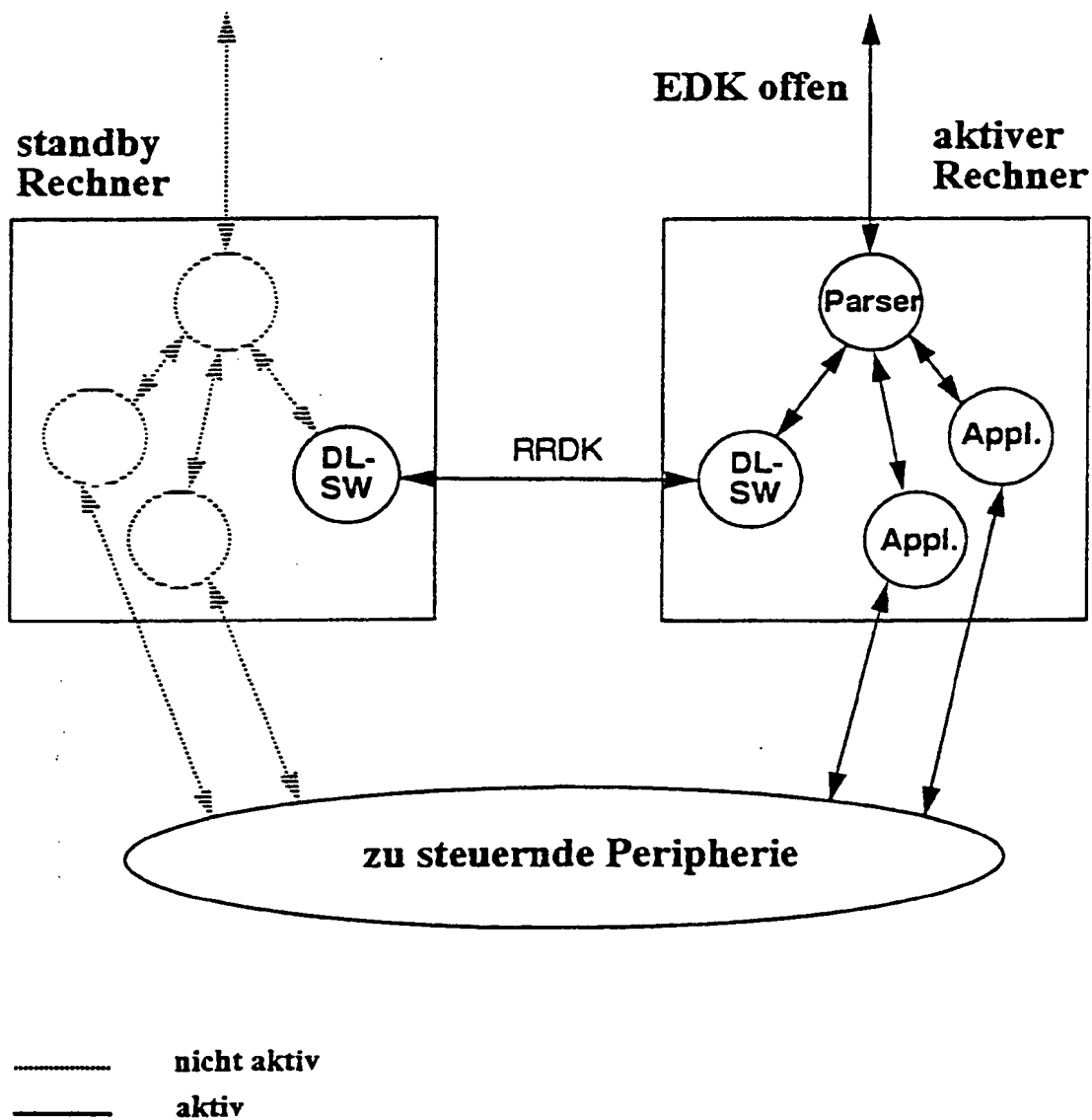


FIG. 2

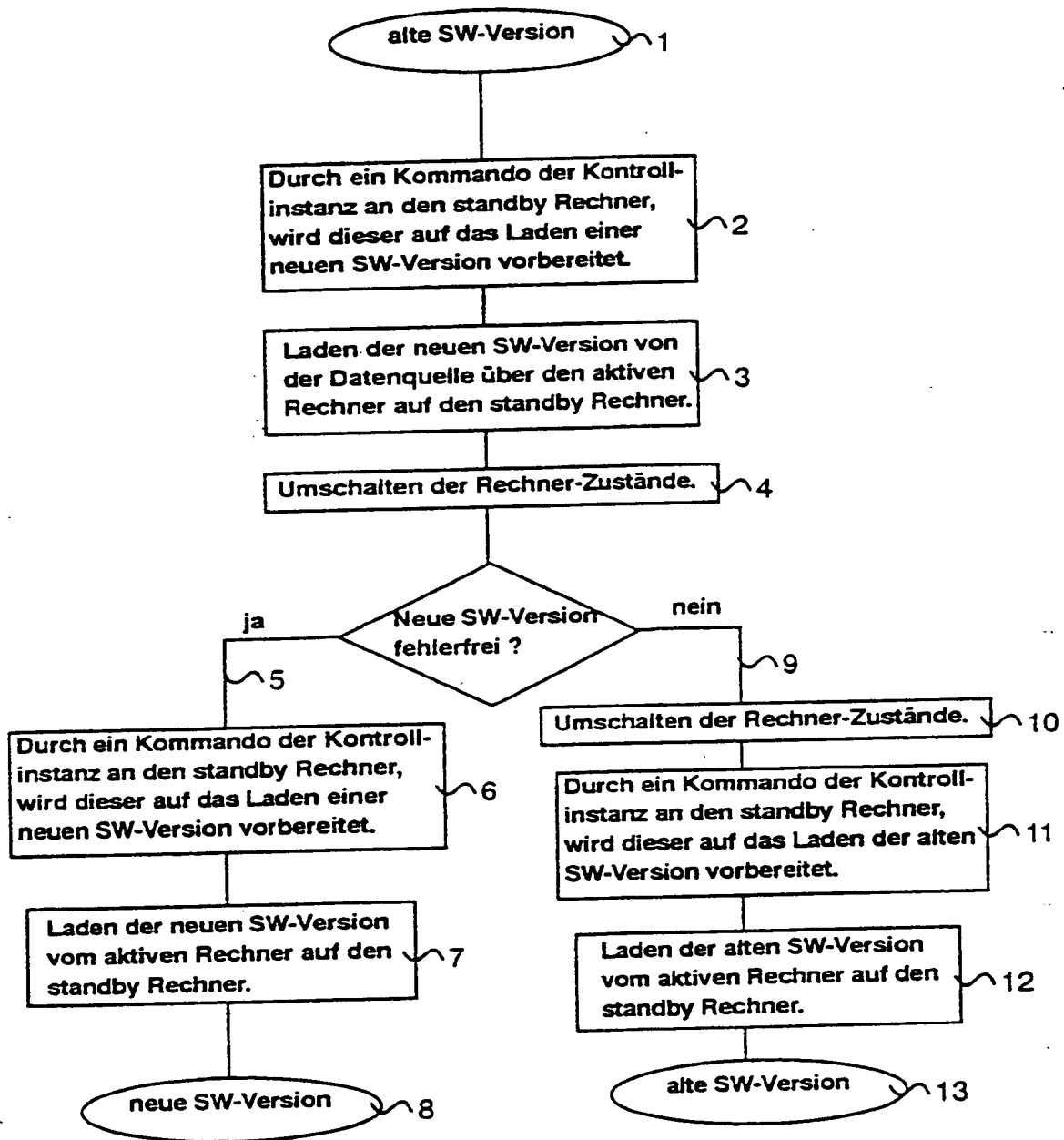


FIG. 3

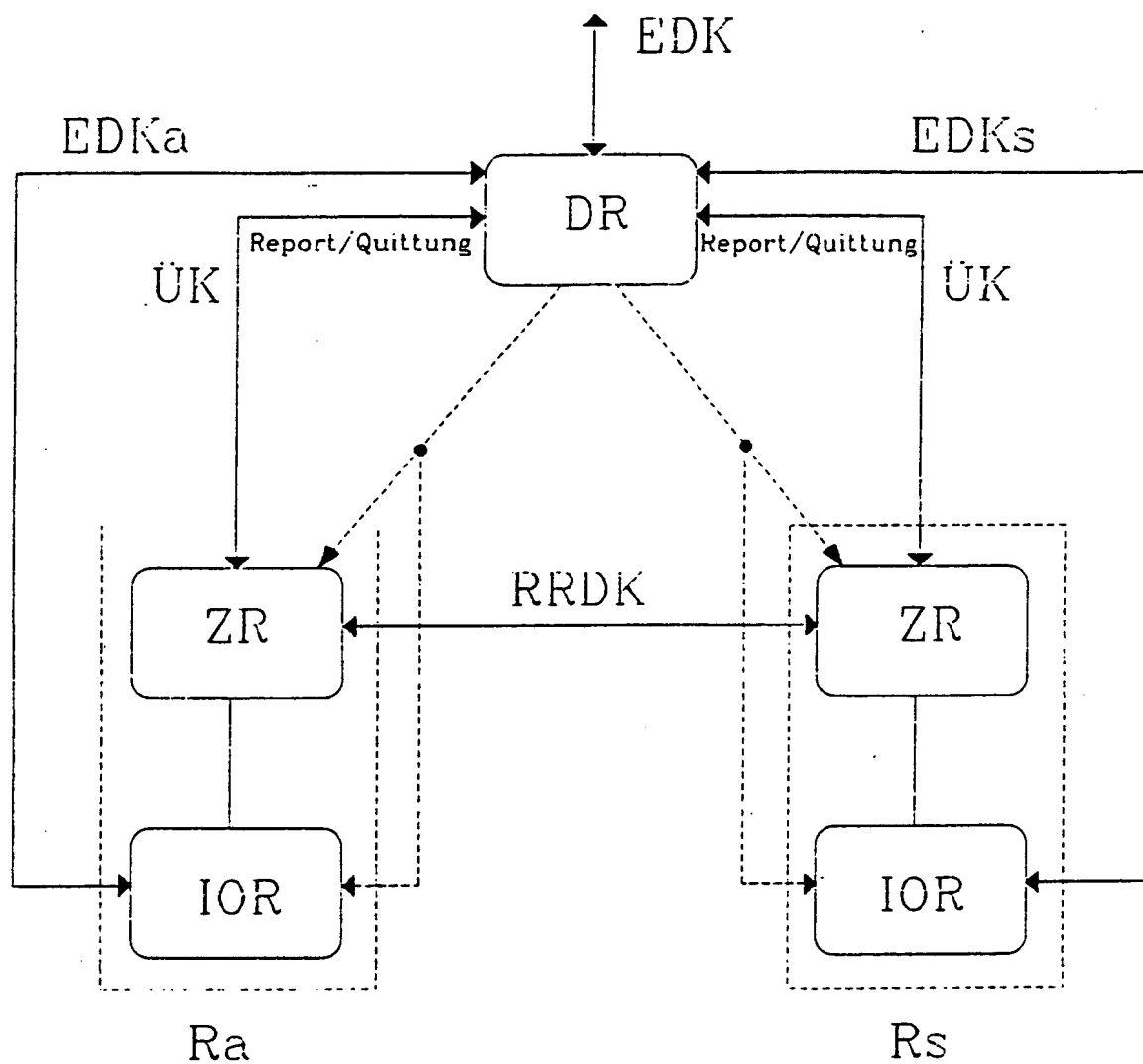


Fig.4



12

Gebrauchsmuster

U1

- (11) Rollennummer G 93 00 562.8
- (51) Hauptklasse H04Q 3/54
- (22) Anmeldetag 18.01.93
- (47) Eintragungstag 04.03.93
- (43) Bekanntmachung
im Patentblatt 15.04.93
- (30) Pri 27.08.92 DE 92 11 546.2
- (54) Bezeichnung des Gegenstandes
Steuerungssystem eines Vermittlungssystems
- (71) Name und Wohnsitz des Inhabers
Siemens AG, 8000 München, DE

32 6 1 3 4 7 DE 0 1

1 Siemens Aktiengesellschaft

Steuerungssystem eines Vermittlungssystems

5 Die Leistungsfähigkeit und Weiterentwicklungsfähigkeit bzw. Veränderbarkeit eines Vermittlungssystems sind in starkem Maße von der Flexibilität des Steuerungssystems abhängig.

10 Der Erfindung liegt die Aufgabe zu Grunde, ein flexibles Steuerungssystem eines Vermittlungssystems anzugeben.

Diese Aufgabe wird durch die Merkmale des Anspruchs 1 gelöst.

15

Mit Hilfe des erfindungsgemäßen Dienstemanagementsystems können die Dienstleistungsmodule einen Dienst bzw. genauer gesagt einen Kommunikationspfad zu einem Dienst nunmehr in einfacher Weise durch Angabe des Namens des Dienstes anfordern, ohne seinen Ort innerhalb des Steuerungssystems zu kennen. Eine Kommunikationsbeziehung zwischen einem aufrufenden Dienst und einem aufgerufenen Dienst wird somit dynamisch aufgebaut. Damit ist die Grundlage für eine unproblematische Veränderbarkeit der Konfiguration des Steuerungssystems während des Betriebs geschaffen.

Eine Ausgestaltung der Erfindung ist durch Anspruch 2 angegeben. Durch diese Ausgestaltung ist es möglich, das Steuerungssystem dynamisch aufzurüsten oder zu erweitern. 30 Des weiteren können bei dieser Ausgestaltung alte Dienstleistungsmodule dynamisch aus dem System entfernt werden, indem die Schritte zur Installation neuer Dienstleistungsmodule in umgekehrter Richtung angewandt werden.

35 Eine weitere Ausgestaltung der Erfindung ist durch Anspruch 3 angegeben. Durch diese Ausgestaltung ist es möglich, die örtliche Verteilung der installierten Dienstleistungsmodu-

1
le dynamisch zu verändern, indem die Dienstleistungsmodule
auf die neuen Orte vervielfältigt werden, während die
Dienstleistungsmodule auf dem ursprünglichen Ort außer
5 Betrieb genommen werden.

Eine weitere Ausgestaltung der Erfindung ist durch An-
spruch 4 angegeben. Bei dieser Ausgestaltung wird zum
einen eine gleichmäßigere Verteilung der Last zwischen den
10 Verarbeitungsplattformen des Steuerungssystems erreicht
und zum anderen eine größere Unabhängigkeit bezüglich des
Ausfalls bzw. Außerbetriebnehmen eines Dienstleistungs-
moduls erreicht.

15 Eine weitere Ausführungsform der Erfindung ist durch
Anspruch 5 angegeben. Durch diese Ausführungsform wird
eine asynchrone Kommunikation zwischen dem aufrufenden
Dienstleistungsmodul und dem aufgerufenen Dienstleistungs-
modul ermöglicht.

20 Eine weitere Ausführungsform der Erfindung ist durch
Anspruch 6 angegeben. Durch diese Ausführungsform wird
eine synchrone Kommunikation zwischen dem aufrufenden
Dienstleistungsmodul und dem aufgerufenen Dienstleistungs-
25 modul ermöglicht.

Eine weitere Ausgestaltung der Erfindung ist durch An-
spruch 7 angegeben. Durch die für eine Dienstleistungs-
kapsel mögliche individuelle Zuteilung von Betriebsmitteln
30 kann erreicht werden, daß die Betriebsmittel für solche
Funktionen, die für die Integrität und den Betrieb des Sy-
stems notwendig sind, in ausreichendem Maße gegenüber sol-
chen Funktionen reserviert werden, die weniger dringend
sind und viele Betriebsmittel benötigen (z.B. Verkehrsmessung).

1

Im folgenden wird die Erfindung anhand der Zeichnung näher erläutert.

5

FIG 1 zeigt ein Schalenmodell für Verarbeitungsplattformen mit System- und Anwendungs-Software,

FIG 2 zeigt die Anforderung eines bestimmten Dienstes durch einen entsprechenden Aufruf an das Dienstemanagementsystem des Betriebssystems,

10

FIG 3 zeigt die Kommunikation zwischen zwei Dienstinstanzen mit Hilfe von Meldungen,

FIG 4 zeigt die Kommunikation zwischen zwei Dienstinstanzen mit Hilfe von Remote-Prozedur-Aufrufen,

FIG 5 zeigt die Dienstetabelle sowie den Ablauf zur Veränderung der Dienstetabelle.

15

FIG 1 zeigt ein Schalenmodell für Verarbeitungsplattformen eines Steuerungssystems, das folgende Schalen aufweist:

Eine erste Schale, die der Hardware der betrachteten drei Verarbeitungsplattformen entspricht,

20

einer zweiten Schale, die der Firmware (Urlader, Hardware-Schnittstelle, Fehlererkennung, usw.) der Verarbeitungsplattformen entspricht,

eine dritte Schale, die der Betriebssystem-Software (Betriebssystemkern, Fehlerbehandlung) entspricht,

25

eine vierte Schale, die der Basis-Software (Datenbasis, Überlaststeuerung usw.) der Verarbeitungsplattform entspricht und die mehrere Dienstleistungskapseln DLK umfaßt,

eine fünfte Schale, die der System-Software (Konfigurations-Software, Recovery-Software usw.) der Verarbeitungsplattform entspricht und die ebenfalls mehrere Dienstleistungskapseln DLK umfaßt,

30

eine sechste Schale, die der Anwender-Software (Vermittlungstechnik, ATM-Cross-Connect, Protokollabwicklung usw.) der Verarbeitungsplattform entspricht, und die ebenfalls

35

mehrere Dienstleistungskapseln DLK umfaßt.

1

4

5

Nur die oberhalb der zweiten Schale ablaufende Software wird in Dienstleistungsmoduln SPU zusammengefaßt und in Dienstleistungskapseln DLK gekapselt. Die Schalen 1 bis 3 stellen zusammen die Grundverarbeitungsplattform dar und sind für jeden Verarbeitungsplattformtyp gleich.

10

15

20

25

30

35

Mit dem Dienstleistungsmodul steht dem Entwickler ein größerer und zweckmäßigerer Container als der heutige Prozess zur Verfügung. Ein Dienstleistungsmodul ist eine Menge von Modulen und Prozessen, die eine starke funktionale Beziehung zueinander haben und zusammen eine Gruppe von Diensten für die übrige Software oder den Teilnehmer des Vermittlungssystems zur Verfügung stellen. Seine Komponenten (Module und Prozesse) befinden sich garantiert im selben Adressraum des Speichers. Deshalb können für die Kommunikation innerhalb eines Dienstleistungsmoduls lokale Prozeduraufrufe und gemeinsame Speichermechanismen benutzt werden.

Das Dienstleistungsmodul ist eine relozierbare Einheit, d.h. es kann von einer Dienstleistungskapsel zu einer anderen Dienstleistungskapsel und ebenso von einer Verarbeitungsplattform zu einer anderen Verarbeitungsplattform zur Produktionszeit des Steuerungssystems bewegt werden. Daraus ergibt sich, daß die Schnittstellen eines Dienstleistungsmoduls durch Ausdrücke der Inter-Prozeß-Kommunikation definiert werden müssen (z.B. CAST, Remote-Prozeduren). Jede Aktion oder jeder Steuerungsdatenfluß zwischen zwei Dienstleistungsmodulen wird durch die Übergabe von Meldungen ausgelöst. Die Kommunikation zwischen zwei Dienstleistungsmodulen erfolgt somit ausschließlich über das Betriebssystem. Darüber hinaus ist das Design und die Implementation eines Dienstleistungsmoduls "location transparent", d.h. ein Dienstleistungsmodul braucht keine

Kenntnis davon zu haben auf welcher Verarbeitungsplattform es abläuft und auf welchen Verarbeitungsplattformen eventuelle Kommunikationspartner laufen.

Ein Dienstleistungsmodul kann weiterhin vervielfältigt werden, so daß dasselbe Dienstleistungsmodul sowohl auf mehreren Steuerungsprozessoren als auch mehrmals auf einem einzigen Steuerungsprozessor auftreten kann. Die Vervielfältigung kann sowohl zum Produktionszeitpunkt des Systems stattfinden oder sogar während der Laufzeit des Systems. Durch die genannte Vervielfältigbarkeit eines Dienstleistungsmoduls wird die Flexibilität des Systems bezüglich der Lastverteilung und der Erweiterbarkeit unterstützt.

Das Dienstleistungsmodul spielt innerhalb des Systems zwei unterschiedliche Rollen. Es kann zum einen Dienstleistungen für andere Dienstleistungsmodule ausführen, oder zum anderen Dienstleistungen von anderen Dienstleistungsmodulen anfordern. In dem erstgenannten Fall handelt es als Server, während es im zweitgenannten Fall als Client handelt. Einige Dienstleistungsmodule können ausschließlich als Server handeln, während andere wiederum als ausschließliche Klienten handeln. Im allgemeinen jedoch spielt ein Dienstleistungsmodul beide Rollen.

Damit ein Dienstleistungsmodul die Dienstleistungen eines anderen Dienstleistungsmodul in Anspruch nehmen kann, muß es wissen, welche Dienstleistungen von dem Server-Dienstleistungsmodul zur Verfügung gestellt werden. Zu diesem Zweck stellt jedes Dienstleistungsmodul sogenannte Dienstschnittstellen zur Verfügung, in der die Dienste beschrieben sind, die das Dienstleistungsmodul innerhalb des Systems zur Verfügung stellt. Diese Dienstschnittstellen werden, nachdem ein Dienstleistungsmodul in

das System geladen worden ist, gegenüber dem Betriebs-
system bekannt gemacht, so daß andere Dienstleistungs-
module die Dienstleistungen des geladenen Dienstleistungs-
moduls über das Betriebssystem ab diesem Zeitpunkt in
Anspruch nehmen können.

Eine Dienstschnittstelle beinhaltet den Namen und Typ des
Dienstes, eine Gruppe von Remote-Prozeduren und Remote-
Puffern, sowie weitere den zugehörigen Dienst spezifi-
zierende Datenobjekte, z.B. Partition.

Das Dienstleistungsmodul stellt einen eigenständigen Bau-
stein zur Systemerstellung dar, der über seine Dienst-
schnittstellen von anderen Dienstleistungsmodulen aufge-
rufen werden kann. Nach der Installierung dieses Bausteins
innerhalb des Systems kann man ihn auf verschiedene Art
mit anderen Dienstleistungsmodulen zusammenfassen, um neue
Systemleistungen bereitzustellen. Dieses Prinzip ist auch
die Basis für Standardbausteine zur Software-Erstellung,
die der momentanen Standardisierung intelligenter Netze
entsprechen.

Im folgenden wird ein weiterer Software-Container,
nämlich eine Dienstleistungskapsel DLK näher erläutert.

Die Dienstleistungskapsel umfaßt eine Gruppe von Dienst-
leistungsmodulen, denen gemeinsam ein bestimmtes Budget von
Betriebsmitteln (z.B. geschützter Adressraum, zwei Spei-
cher, Zeitgeber) zur Verfügung gestellt ist. Eine Dienst-
leistungskapsel kann on-line, d.h. ohne Betriebsunterbre-
chung, in das System eingebracht oder durch eine aufgerü-
stete Version ersetzt werden.

Durch das Zusammenfassen von Dienstleistungsmodulen in

1

7

Dienstleistungskapseln lassen sich neue Systeme oder kundenspezifische Leistungsmerkmale schaffen. Darüber hinaus ist es möglich, bestehenden Dienstleistungskapseln geänderte oder vollständig neue Dienstleistungsmodul-
5 hinzuzufügen. Die Tatsache, daß Dienstschnittstellen gewartet werden können, ermöglicht das Erstellen neuer Kombinationen zu niedrigen Kosten und in kürzerer Zeit. Darüber hinaus können die Prüfung und Wartung eines Dienstleistungsmoduls als isolierbare Funktionseinheiten ausge-
10 führt werden. Dadurch werden übergreifende Funktionstests in Umfang und Komplexität vermindert, was die Entwicklungskosten weiter reduziert.

15 Das Aufrüsten von Betriebsvermittlungsstellen durch Hinzufügen von Dienstleistungskapseln ist ein Mittel, Entwicklungskosten zu vermindern und dadurch Wettbewerbsvorteile zu erhalten. Auf diese Weise kann der Netzbetreiber neue Leistungsmerkmale mit verringerter Logistik schnell
20 in das Netz einbringen. Die neuen Merkmale lassen sich in bestehenden oder neuen Verarbeitungsplattformen hinzufügen. In beiden Fällen sind folgende Schritte erforderlich:

25 In einem ersten Schritt werden die Dienstleistungskapsel off-line durch Zusammenfassen mehrerer Dienstleistungsmodul erstellt und anschließend an die entsprechenden Vermittlungsstellen ausgeliefert.

In einem zweiten Schritt werden in einer entsprechenden Vermittlungsstelle on-line neue Verarbeitungsplattform-
30 typen erstellt.

In einem dritten Schritt vollzieht sich das Laden neuer Kapseln in bestehende Verarbeitungsplattformen, um neue Verarbeitungsplattformtypen zu vervollständigen, oder das
35 Laden vollständiger Verarbeitungsplattformtypen in neue

Verarbeitungsplattformen.

In einem vierten Schritt vollzieht sich schließlich das Aktivieren neuer Kapseln oder eines neuen

5 Verarbeitungsplattformtyps auf der bestehenden bzw. neuen Verarbeitungsplattform.

10 Durch die genannten Schritte kann eine neue Kombination bestehender Funktionen ohne Betriebsstörung in das System eingefügt werden. Neue Leistungsmerkmal-Software läßt sich auf dieselbe Weise einbringen, weil das Betriebssystem jede Dienstleistungskapsel als eigene Verarbeitungseinheit behandelt.

15 Wie bereits vorher erwähnt, werden Kommunikationsbeziehungen zwischen den Dienstleistungsmodulen dynamisch hergestellt. Diese Funktion wird im folgenden als Dienstadressierung bezeichnet und durch ein Dienstemanagementsystem innerhalb der Betriebssystem-Software zur Verfügung gestellt.

25 Jede Implementierung eines bestimmten Dienstes innerhalb eines Dienstleistungsmoduls wird als Instanz eines bestimmten Dienstes (Diensttyp) oder abgekürzt einfach als Dienstinstanz bezeichnet. Die von einem Dienstleistungsmodul durch entsprechende Dienstinstanzen zur Verfügung gestellten Dienstleistungen werden dem System in bereits erwähnten Dienstschnittstellen bekanntgegeben. Dabei liegt für jede Dienstinstanz eine entsprechende Dienstschnittstelle vor, die von dem jeweiligen Dienstleistungsmodul exportiert und in einer Spezifikation (Kurzerklärung mit Syntax und Semantik) erläutert wird.

35 Immer wenn eine Dienstinstanz eines Dienstleistungsmoduls eine Kommunikationsbeziehung mit einer anderen Dienst-

stanz eines anderen Dienstleistungsmoduls zwecks Ausführung eines Dienstes herstellen will, so führt sie über das Betriebssystem einen Aufruf an das Dienstmanagementsystem durch, in welchem der Name des Typs des angeforderten Dienstes und gegebenenfalls zusätzliche Kriterien für die Auswahl eines Server-Dienstleistungsmoduls angegeben werden.

Mögliche zusätzliche Auswahlkriterien sind

- eigene Verarbeitungsplattform, d.h. geringster dynamischer Aufwand,
- Partitionen, die den Dienst auf einen bestimmten Datenbereich oder Ressourcen-Bereich eingrenzen,
- explizite Angabe der Verarbeitungsplattform und/oder der Dienstleistungskapsel.

Als Resultat des Aufrufs erhält die aufrufende Kunden-Dienstinstanz einen sogenannten Kommunikationspfad zurück, mit Hilfe dessen die Kommunikation zu der ausgewählten Server-Dienstinstanz aufgenommen werden kann. Bei dem Kommunikationspfad handelt es sich um ein Datenfeld, das im wesentlichen Remote-Referenzen (RPC's und CAST) zu der ausgewählten Server-Dienstinstanz enthält. Die eigentliche Kommunikation wird sodann durch Senden von Meldungen (Cast) oder Aufrufen von Remote-Prozeduren (RPC) durchgeführt.

FIG 2 zeigt die Anforderung eines bestimmten Dienstes durch den Aufruf "Connect-Service" an das Dienst-Management System eines Betriebssystems BS. In Fig. 2 erfolgt der genannte Aufruf innerhalb einer Verarbeitungsplattform GP3 eines Steuerungssystems, das noch zwei weitere Verarbeitungsplattformen GP1 und GP2 aufweist. Als Resultat des genannten Aufrufs erhält die aufrufende Dienst-

1
10
15
20
25
30
35

Instanz des Dienstleistungsmoduls SPU0 eine Referenz auf die Dienstschnittstelle einer Dienstinstanz, die den angeforderten Dienst ausführen kann. Die genannte Dienstschnittstelle enthält u.a. eine Sammlung von Remote-Referenzen, mit deren Hilfe in Form von Meldungen (Cast) und Remote-Prozedur-Aufrufen (RPC) die Kommunikation mit der vom Dienstemanagement ausgewählten Dienstinstanz durchgeführt werden kann.

Bei der Auswahl einer geeigneten Dienstinstanz berücksichtigt das Dienstemanagementsystem Last-/und Maintenance-Zustände derjenigen Plattformen, auf denen sich geeignete Dienstinstanzen befinden. Die Informationen über die genannten Plattformzustände entnimmt das Dienstmanagementsystem einer eigenen lokalen Zustandstabelle, die durch das Überlastbehandlungssystem und das Maintenance-System regelmäßig aufdatiert wird.

FIG 3 zeigt die Kommunikation zwischen zwei Dienstinstanzen mit Hilfe von Meldungen. Bei dem Steuerungssystem auf der linken Seite von FIG 3 befindet sich die anfordernde Dienstinstanz in dem Dienstleistungsmodul SPU0 des Steuerungsprozessors GP3 und die ausgewählte Dienstinstanz innerhalb des Dienstleistungsmoduls SPUL desselben Steuerungssprozessors. Bei dem Steuerungssystem auf der rechten Seite von FIG 3 befindet sich die anfordernde Dienstinstanz in dem Dienstleistungsmodul SPU0 des Steuerungsprozessors GP3 und die ausgewählte Dienstinstanz in dem Dienstleistungsmodul SPUL des Steuerungsprozessors GP1.

FIG 4 zeigt die Kommunikation zwischen zwei Dienstinstanzen mit Hilfe von Remote-Prozedur-Aufrufen (RPC). Die örtliche Lage der anfordernden Dienstinstanz und der angeforderten Dienstinstanz innerhalb der beiden dargestellten

Steuerungssysteme ist zu der in FIG 3 bereits angegebenen örtlichen Lage identisch.

- 5 Eine Client-Dienstinstanz und eine Server-Dienstinstanz sind in der Lage, Statusinformationen über ihren Kommunikationsablauf zu speichern, da anderenfalls jeder Kommunikationsvorgang zwischen den beiden durch einen erneuten Aufruf "Connect-Service" eingeleitet werden müßte. Es kann
10 nun Situationen geben, in denen zum Zeitpunkt eines Ausfalls oder einer Außerbetriebnahme der Server-Dienstinstanz wegen einer Wartung die Kommunikationsbeziehung zwischen der Server-Dienstinstanz und der Client-Dienstinstanz noch bestehen kann. Versucht in einem solchen Fall
15 die Client-Dienstinstanz einen weiteren Kommunikationsvorgang, z.B. durch Senden von Meldungen zur Server-Dienstinstanz durchzuführen, so erhält sie von der Inter-Prozeß-Kommunikation eine entsprechende Fehlermeldung zurück. Dies bedeutet, daß die Kommunikationsbeziehung von seiten
20 der Server-Dienstinstanz aufgelöst worden ist. Die Client-Dienstinstanz muß nun erneut eine Kommunikationsbeziehung mit einer geeigneten Server-Dienstinstanz aufbauen, um die Durchführung des Dienstes fortzusetzen. Die genannte Fehlermeldung der Inter-Prozeß-Kommunikation enthält eine
25 Angabe darüber, ob es sich um eine Nichtverfügbarkeit der Server-Dienstinstanz (wegen Wartung oder Überlast) oder um einen Fehler des angesprochenen Puffers (z.B. wegen Abschluß einer sanften Außerbetriebnahme) handelt.
- 30 Im Falle eines sanften Außerbetriebnehmens eines Dienstleistungsmoduls wegen Erweiterung können alte bestehende Kommunikationsbeziehungen während eines Überspannungszeitraumes erhalten bleiben, während eine Client-Dienstinstanz zur Aufnahme einer erstmaligen Kommunikationsbeziehung
35 durch den Aufruf "Connect-Service" außer dem Kommunikationspfad als optionalen Parameter ein sogenanntes Erwei-

terungszeichen zurückerhält. Das genannte Erweiterungs-
zeichen ist dann Teil des Nachrichten-Headers jeder
Nachricht in der Inter-Prozeß-Kommunikation. Bei der
5 genannten sanften Erweiterung sind während des Erwei-
terungszeitraums somit sowohl die alte Dienstinstanz
als auch die neue Dienstinstanz verfügbar und das genannte
Erweiterungszeichen wird benützt, um anzuzeigen, ob mit
der alten Server-Dienstinstanz oder der neuen Server-
10 Dienstinstanz kommuniziert werden soll.

Aus funktionaler Sicht kann unter den Dienstinstanzen zwi-
schen vervielfältigten Dienstinstanzen eines Dienstes und
partitionierten Dienstinstanzen eines Dienstes unter-
15 schieden werden. Vervielfältigte Dienstinstanzen entstehen je-
weils durch die Vervielfältigung eines gesamten Dienstlei-
stungsmoduls. Partitionierte Dienstinstanzen eines
Dienstes weisen denselben Dienstschnittstellentyp wie
vervielfältigte Dienstinstanzen desselben Dienstes auf,
20 führen den Dienst jedoch für einen unterschiedlichen
Bereich von Daten oder eine unterschiedliche Menge von
Ressourcen durch (z.B. kann ein "File-Dienst" für einen
Halbleiterspeicher oder einen Magnetspeicher ausgeführt
werden). Die Server-Dienstinstanzen weisen eine sog.
25 Partitionskennung auf, um zwischen partitionierten
Dienstinstanzen unterscheiden zu können.

Vervielfältigte Dienstinstanzen weisen denselben Dienst-
schnittstellentyp auf und besitzen dieselbe Partitions-
30 kennung, da sie den entsprechenden Dienst für denselben
Bereich von Daten oder dieselbe Menge von Ressourcen
durchführen. Vervielfältigte Dienstinstanzen können auf
unterschiedlichen Verarbeitungsplattformen vorhanden sein
und eine identische oder unterschiedliche Implementation
35 aufweisen. Das Dienstemanagement benützt für die Auswahl
einer Dienstinstanz einen bestimmten Auswahl-Algorithmus,
der Lastzustände und Recoveryzustände der Verarbeitungs-

1

13

plattformen berücksichtigt, um eine optimale Lastverteilung und Fehlertoleranz zu unterstützen. Diese Auswahl-Algorithmen sind für die anforderndenden Dienstinstanzen nicht sichtbar. Die Client-Dienstinstanz braucht somit zwischen mehreren vervielfältigten Dienstinstanzen desselben Dienstes nicht zu unterscheiden.

10

Aus der Sichtweise der Programmiersprache handelt es sich bei einer Dienstschnittstelle um einen speziellen Datentyp, durch den ein Dienst bzw. genauer gesagt ein bestimmter Diensttyp beschrieben wird. Der genannte Datentyp enthält u.a. eine vom Diensttyp abhängige Sammlung von Remote-Referenzen (Remote-Prozedur-Aufrufe RPC und Puffer), die im folgenden auch als Kommunikationspfad bezeichnet wird. Jede Deklaration einer Variable des Datentyps "Diensttyp" entspricht der Erzeugung einer Dienstinstanz. Diese Erzeugung umfaßt die statische Initialisierung der genannten Remote-Referenzen.

20

Eine Dienstinstanz ist innerhalb des Systems durch eine Diensttypkennung, eine Partitionskennung, eine Kapselkennung und eine Dienstleistungsmodulkennung eindeutig spezifiziert.

25

Die Diensttypkennung wird durch die Support-Software, genauer gesagt durch den sogenannten Systembinder (Linker) während der Produktionszeit oder Laufzeit des Systems erzeugt. Eine solche Diensttypkennung wird jeder deklarierten Dienstvariablen zugeordnet. Aufgrund der Zuordnung ist die deklarierte Dienstvariable nun innerhalb des Systems als Instanz eines bestimmten Dienstes bekannt. Alle Dienstinstanzen, deren Variablentyp (Diensttyp) und Variablen-Name gleich ist, erhalten innerhalb des Systems dieselbe Diensttypkennung. Eine Instanz eines bestimmten

30

35

920012

Dienstes kann seine Diensttypkennung niemals ändern.

Die Partitionskennung einer Dienstinstanz wird ebenfalls zur Produktionszeit (statisches Bekanntmachen) oder zur Laufzeit (dynamisches Bekanntmachen) erzeugt. Der Typ der Partitionskennung ist ein Bereich von Daten oder eine Menge von Ressourcen und wird innerhalb der ersten Komponente des Datentyps "Diensttyp" definiert.

Die Partitionskennung ermöglicht es der Anwender-Dienstinstanz zwischen mehreren Dienstinstanzen zu unterscheiden. Alle Dienstinstanzen eines bestimmten Dienstes, die dieselbe Partitionskennung aufweisen, stellen aus der Sichtweise des Dienstemanagementsystems vervielfältigte Dienstinstanzen dar. Alle Dienstinstanzen eines bestimmten Dienstes mit unterschiedlichen Partitionskennungen stellen aus der Sichtweise des Dienstemanagementsystems partitionierte Dienstinstanzen dar.

Es ist möglich, eine vorher zugeordnete Partitionskennung für eine Dienstinstanz während der Laufzeit zu ändern. Dies ist allerdings nur für solche Dienstinstanzen möglich, deren Zuordnung einer Partitionskennung auf dynamische Weise (dynamisches Bekanntmachen) erfolgt ist.

Die Kapselkennung und die Dienstleistungsmodulkennung beschreiben die Lage einer Dienstinstanz innerhalb des gesamten Systems. Jede Dienstinstanz hat innerhalb des Systems somit eine eindeutige örtliche Lage. Es ist nicht erlaubt, dieselbe Instanz eines Dienstes innerhalb eines Dienstleistungsmoduls mehrmals zu erzeugen (Restriktion für den Compiler). Es ist allerdings erlaubt, einer bestimmten Dienstinstanz eine zusätzliche Partitionskennung zuzuordnen. Die genannte Restriktion bedeutet

13.01.90

1

15

also, daß es nicht möglich ist, mehr als eine Dienstinstanz eines bestimmten Dienstes innerhalb eines bestimmten Dienstleistungsmoduls zu haben.

5

10

Die Dienstleistungsmodulkennung wird durch den Linker erzeugt und dem off-line-Systembauer übergeben. Die Kapselkennung wird durch den off-line-Systembauer erzeugt. Die Kapselkennung und die Dienstleistungsmodulkennung werden vom off-line-Systembauer zum on-line-Systembauer in Form von Tabellen übergeben. Diese Tabellen werden vom on-line-Systembauer benützt, wenn er eine sogenannte Dienstetabelle erzeugt.

15

20

Jede Dienstinstanz hat innerhalb des Systems eine eindeutige örtliche Lage. Diese örtliche Lage wird durch eine Ortskennung (Kapselkennung und Dienstleistungsmodulkennung) beschrieben. Das Dienstemanagement erlaubt es, daß mehrere Dienstinstanzen eines bestimmten Dienstes, d.h. eines bestimmten Dienstschnittstellentyps dieselbe Ortskennung besitzen, solange sie nicht dieselbe Partitionskennung aufweisen.

25

30

35

Eine Dienstinstanz kann durch einen Remote-Prozedur-Aufruf RPC oder eine Nachricht (Cast) in einem der Puffer der genannten Dienstinstanz angerufen werden. Der Aufruf einer Dienstinstanz eines bestimmten Dienstes über die Grenzen einer Dienstleistungskapsel hinaus ist durch ein entsprechendes Feld von Remote-Referenzen (Puffer-Referenzen und RPC-Referenzen) und den Gebrauch von entsprechenden Kommunikationseinrichtungen des Betriebssystems implementiert. Der Aufruf einer Instanz eines bestimmten Dienstes innerhalb einer Dienstleistungskapsel ist durch den Gebrauch eines Feldes auf Local-Referenzen (Zeiger auf Puffer und Prozeduren) implementiert. Die Größe eines solchen Feldes hängt von der Anzahl der Komponenten eines Dienstes ab.

1

16

Die Aufgabe des Dienstmanagementsystems ist es aufgrund einer Anforderung eines Dienstes eine entsprechende Dienstinstanz zur Durchführung des angeforderten Dienstes auszuwählen und deren Kommunikationspfad (Feld von Referenzen) der anfordernden Dienstinstanz zu übergeben. Die genannte Anforderung eines Dienstes wird mit Hilfe des Betriebssystem-Aufrufs "Connect Service" bewirkt. Der genannte Betriebssystem-Aufruf ermittelt den genannten Kommunikationspfad, indem er in verschiedenen Tabellen nachsieht. Die in diesem Zusammenhang wichtigsten Tabellen sind eine statische Dienstetabelle und eine transiente Dienstetabelle. Die Elemente dieser Tabellen sind die Dienstkennungen, die Kapselkennungen, die Dienstleistungsmodulkennungen, die Partitionskennungen und schließlich die Kommunikationspfade.

Die Diensttypkennung ist als ein Parameter innerhalb des Betriebssystems-Aufrufs "Connect Service" unbedingt nötig. Andere Parameter, wie z.B. die Partitionskennung oder die Kennungen bezüglich der örtlichen Lage (Kapselkennung und Dienstleistungsmodulkennung) sind optionale Parameter. Wenn der genannte Betriebssystem-Aufruf erfolgreich ist, wird - wie bereits erwähnt - ein Datenfeld mit Remote-Referenzen (Kommunikationspfad), der aufrufenden Dienstinstanz zurückgegeben.

Das Dienstmanagementsystem benützt zur Durchführung seiner Aufgabe die genannten Dienstabellen, welche Informationen über die verfügbaren Dienste des Systems, die zur Durchführung der Dienste vorhandenen Dienstinstanzen und die zu den Dienstinstanzen zugehörigen Kommunikationspfade beinhalten. Diese Dienstabellen werden durch Werkzeuge (Tools) der Support-Software erzeugt, die sowohl off-line als auch on-line arbeiten.

1

17

Die erzeugten Dienstabellen enthalten Informationen über die verschiedenen Verarbeitungsplattformen des Steuerungssystems und die verfügbaren Dienstinstanzen auf den vorhandenen Verarbeitungsplattformen.

5

Insbesondere die Ortskennungen werden durch ein On-Line-Werkzeug erzeugt. Die Ortskennungen werden von der Inter-Prozess-Kommunikation und der Inter-Prozessor-Kommunikation des Betriebssystems benutzt. Das Dienstemanagementsystem macht keine weiteren Annahmen über die erzeugten Ortskennungen. Es geht davon aus, daß die erzeugten Ortskennungen gültig sind und bietet der Inter-Prozess-Kommunikation, der Inter-Prozessor-Kommunikation und dem internen Transportprotokoll anhand der Ortskennungen die von diesen Systemen benötigten Informationen über die Zielbestimmungen an.

10

15

20

Um die genannten Dienstabellen zu erzeugen, ist eine Informationsweitergabe über verschiedene Werkzeuge der Support-Software notwendig, die im folgenden näher erläutert wird.

25

30

Der Compiler erkennt bei der Compilierung zunächst die Deklarationen von Dienstvariablen (Dienstinstanzen) und deren Schnittstellenkomponenten (Puffer etc.) innerhalb des Source-Codes. Der Compiler reicht diese Information an den Linker weiter. Nach der Compilierung sind die genannten Dienstinstanzen einem bestimmten Dienstleistungsmodultyp zugeordnet.

35

Der Linker reicht sodann die genannten Informationen an den off-line Systembauer weiter. Nach dem Binden durch den Linker gehören die genannten Dienstinstanzen zu einem bestimmten Dienstleistungskapseltyp. Die Dienstleistungs-

1

5

10

15

20

25

30

35

modulkennung, die die relative Lage des Dienstleistungsmoduls innerhalb einer Dienstleistungskapsel angibt, ist zu diesem Zeitpunkt festgelegt worden. Außerdem ist zu diesem Zeitpunkt auch die relative Position der Komponenten der Schnittstellen der Dienstinstanzen innerhalb einer Dienstleistungskapsel festgelegt worden.

Der off-line-Systembauer ordnet nun den vom Linker überreichten Dienstinstanzen entsprechende Dienstkennungen zu. Außerdem erzeugt er für jede Verarbeitungsplattform einen bestimmten Ladungstyp, durch den jeweils die Funktion einer Verarbeitungsplattform bestimmt wird. Die relative Position einer Dienstleistungskapsel innerhalb eines Ladungstyps ist nun festgelegt.

Der on-line-Systembauer kennt die vom off-line-Systembauer erzeugte Konfiguration der Hardware und der zugehörigen Ladungstypen der Verarbeitungsplattformen. Er erzeugt daraus die notwendigen Tabellen für das Dienstmanagementsystem. Er kennt die Beziehungen zwischen den Ladungstypen und den spezifischen Ortskennungen einer Verarbeitungsplattform. Die von ihm erzeugte Dienstetabelle enthält die Beziehungen zwischen Dienstinstanzen, deren örtlicher Lage und deren zugehörigen Schnittstellenkomponenten.

FIG 5 zeigt den Gebrauch der Dienstetabelle durch das Dienstmanagementsystem.

Das Dienstmanagementsystem DMS enthält gemäß FIG 5 zur Durchführung seiner Aufgabe eine Dienstetabelle ST, in der die Schnittstellen von Instanzen bereits bekanntgemachter Diensttypen abgespeichert sind. Immer wenn eine Dienstinstanz einen Dienst von einem anderen Dienstleistungsmodul anfordern will und noch keine Kommunikationsbeziehung zu

einer entsprechenden Dienstinstanz besteht, übergibt es an das Dienstmanagementsystem den Namen des angeforderten Diensttyps, z.B. den Namen "Dienst 1". Das Dienstmanagementsystem beinhaltet die Namen der bereits bekanntgemachten Dienste (Diensttypen) in einer Dienstenamens-tabelle innerhalb der Dienstetabelle ST, sowie in einer Instanzentabelle die Ortskennungen und Partitionskennungen derjenigen Dienstinstanzen, die den genannten Dienst ausführen. Anhand des vom anfordernden Dienstleistungsmodul übergebenen Namens kann das Dienstmanagementsystem somit aus den möglichen Dienstleistungsmoduln das hinsichtlich einer optimalen Lastverteilung am besten geeignete Dienstleistungsmodul, das den angeforderten Dienst beinhaltet, auswählen und dem anfordernden Dienst den entsprechenden Kommunikationspfad zu einer Dienstinstanz des angeforderten Dienstes übergeben. Die anfordernde Dienstinstanz kann nun die Kommunikation mit der zugehörigen Dienstinstanz aufnehmen.

Anstelle einer bloßen Namensangabe des angeforderten Dienstes kann der anfordernde Dienst auch zusätzliche Auswahlkriterien angeben, z.B. die Angabe eines bestimmten Steuerungsprozessors, die Angabe einer bestimmten Instanz eines bestimmten Dienstes oder einer bestimmten Dienstleistungskapsel.

Ein bestimmter Dienst, z.B. "Dienst 1" ist von mehreren verschiedenen Instanzen durchführbar. Jede Instanz führt den Dienst jedoch nur für einen bestimmten Bereich von Daten oder eine bestimmte Menge von Ressourcen durch. Dieser Umstand wird durch eine Partitionskennung ID_p markiert.

In FIG 5 sind drei Dienstleistungsmodule, nämlich SPUL,

1

20

5

10

15

20

25

30

35

SPU2 und SPU3 in das System eingebracht worden, wobei es sich bei den Dienstleistungsmoduln SPU1 und SPU2 um identische Dienstleistungsmodule handelt und somit wenigstens eines von den beiden genannten Dienstleistungsmoduln durch Vervielfältigung entstanden ist. Aufgrund der Identität der beiden Dienstleistungsmoduln besitzen einander entsprechende Dienste der beiden Dienstleistungsmoduln, z.B. "Dienst 1", auch dieselben Partitionskennungen ID_p.

Nach dem Einbringen eines Dienstleistungsmoduls in das System, sei es durch Laden oder Vervielfältigen, erfolgt das Bekanntmachen der Dienste des neu eingebrachten Dienstleistungsmoduls gegenüber dem Dienstmanagementsystem DMS, dessen Ablauf nun näher beschrieben wird.

In einem ersten Schritt wird aufgrund einer Anforderung "Announce Service", die von einer anfordernden Instanz an das Dienstmanagementsystem gestellt wird, die Ortskennung (Kapselkennung und Dienstleistungsmodulkennung) aus dem Prozeßkontrollblock der aufrufenden Instanz entnommen.

Die Diensttypkennung und die Partitionskennung werden als Parameter in der Anforderung "Announce Service" mitübergeben. Für Dienste, die dynamisch, d.h. zur Laufzeit des Systems bekanntgegeben werden, steht in jedem Dienstleistungsmodul ein separater Prozeß zur Verfügung, der jeweils nur Dienste seines eigenen Dienstleistungsmoduls anmelden kann. Der genannte Prozeß kann pro Dienstinstantz mehrere Partitionskennungen anmelden.

In einem zweiten Schritt wirft das Dienstmanagement nun anhand der übergebenen Diensttypkennung und der Ortskennung einen Blick in die bezüglich des Diensttyps relevante Instanzentabelle innerhalb der Dienstetabelle.

Wenn die in der Anforderung übergebene Partitionskennung in der Instanztabelle bereits definiert ist, geschieht nichts und es wird ein entsprechender Rückgabewert der aufrufenden Instanz zurückgegeben. Wenn in der genannten Instanztabelle jedoch keine korrespondierende Dienstinstanz mit derselben Partitionskennung gefunden wird, wird die Anforderung ausgeführt, d.h. die neue Partitionskennung wird in die Partitionsliste innerhalb der Instanztabelle eingetragen und daraufhin der die Bekanntmachung anfordernden Instanz ein entsprechender Rückgabewert zurückgegeben.

In einem dritten Schritt muß die soeben innerhalb einer bestimmten Verarbeitungsplattform erfolgte Bekanntmachung einer Dienstinstanz auch auf die übrigen Verarbeitungsplattformen verteilt werden. Die Bekanntmachungsinformationen der neuen Dienstinstanz müssen dazu zunächst in einer lokalen Zwischenspeichertabelle abgespeichert werden, bevor sie nach einer angemessenen Zeitspanne durch eine Multicast-Kommunikation zu den Dienstmanagementsystemen der anderen Verarbeitungsplattformen gesandt werden. Wenn die genannte lokale Zwischenspeichertabelle bereits vor der genannten angemessenen Zeitspanne einen bestimmten Anfüllungsgrad überschritten haben sollte, wird ebenfalls eine Multicast-Kommunikation auf Betriebssystemebene ausgelöst.

Zu dem soeben beschriebenen Vorgang des Bekanntmachens einer neuen Dienstinstanz innerhalb des Steuerungssystems existiert ein umgekehrter Vorgang zum Löschen der Bekanntmachung einer neuen Dienstinstanz. Dieser umgekehrte Vorgang wird durch den Betriebssystemaufruf "Withdraw Service" ausgelöst und in analoger Weise zu dem vorher erläuterten Vorgang des Bekanntmachens durchgeführt.

1

Schutzansprüche

1. Steuerungssystem eines Vermittlungssystems mit
- a) wenigstens einem Steuerungsprozessor
 - 5 b) jeweils einem Betriebssystem pro Steuerungsprozessor, das die Betriebsmittel des Vermittlungssystems steuert, verwaltet und überwacht,
 - c) wenigstens einem Dienstleistungsmodul (SPU) pro Steuerungsprozessor, das jeweils wenigstens einen
 - 10 Dienst beinhaltet, das auf mehreren Steuerungsprozessoren und/oder mehrmals auf einem Steuerungsprozessor eingebracht sein kann und das zur Ausführung eines seiner Dienste einen Dienst eines anderen Dienstleistungsmodul anfordern kann,
 - 15 d) einem Dienstemanagementsystem, das jeweils im Betriebssystem enthalten ist und das einem Dienstleistungsmodul, das einen bestimmten Dienst anfordert, den Kommunikationspfad mitteilt, über den die Kommunikation zu einem ausgewählten Dienstleistungsmodul aufgenommen werden kann, das den ange-
 - 20 fordernten Dienst beinhaltet.
2. Steuerungssystem nach Anspruch 1,
- 25 d a d u r c h g e k e n n z e i c h n e t ,
daß ein Dienstleistungsmodul auf mehreren Steuerungsprozessoren und/oder mehrmals auf einem Steuerungsprozessor in das Steuerungssystem einbringbar ist, ohne den Betrieb der Steuerungsprozessoren zu unterbrechen.
- 30 3. Steuerungssystem nach Anspruch 1 oder 2,
d a d u r c h g e k e n n z e i c h n e t ,
daß ein eingebrachtes Dienstleistungsmodul auf mehreren Steuerungsprozessoren und/oder mehrmals auf einem Steuerungsprozessor vervielfältigbar ist, ohne den Betrieb
- 35 der Steuerungsprozessoren zu unterbrechen.

100137

1

23

4. Steuerungssystem nach einem der Ansprüche 1 bis 3,
d a d u r c h g e k e n n z e i c h n e t ,
daß das Dienstemanagementsystem die Auswahl eines Dienst-
5 leitungsmoduls für einen angeforderten Dienst in Abhängig-
keit der Lastzustände der Steuerungsprozessoren durch-
führt.

10

5. Steuerungssystem nach einem der Ansprüche 1 bis 4,
d a d u r c h g e k e n n z e i c h n e t ,
daß die Dienstleistungsmoduln die Kommunikation über den
genannten Kommunikationspfad durch Senden von Meldungen
realisieren.

15

6. Steuerungssystem nach einem der Ansprüche 1 bis 4,
d a d u r c h g e k e n n z e i c h n e t ,
daß die Dienstleistungsmoduln die Kommunikation über den
genannten Kommunikationspfad durch Aufrufen von Remote-
Prozeduren realisieren.

20

7. Steuerungssystem nach einem der Ansprüche 1 bis 6,
d a d u r c h g e k e n n z e i c h n e t ,
daß

25

- a) mehrere Dienstleistungsmoduln zu einer Dienstleistungs-
kapsel zusammengefaßt sind,
- b) einer Dienstleistungskapsel jeweils ein bestimmtes
Budget an Betriebsmitteln zugeordnet ist.

30

35

000000

FIG 1

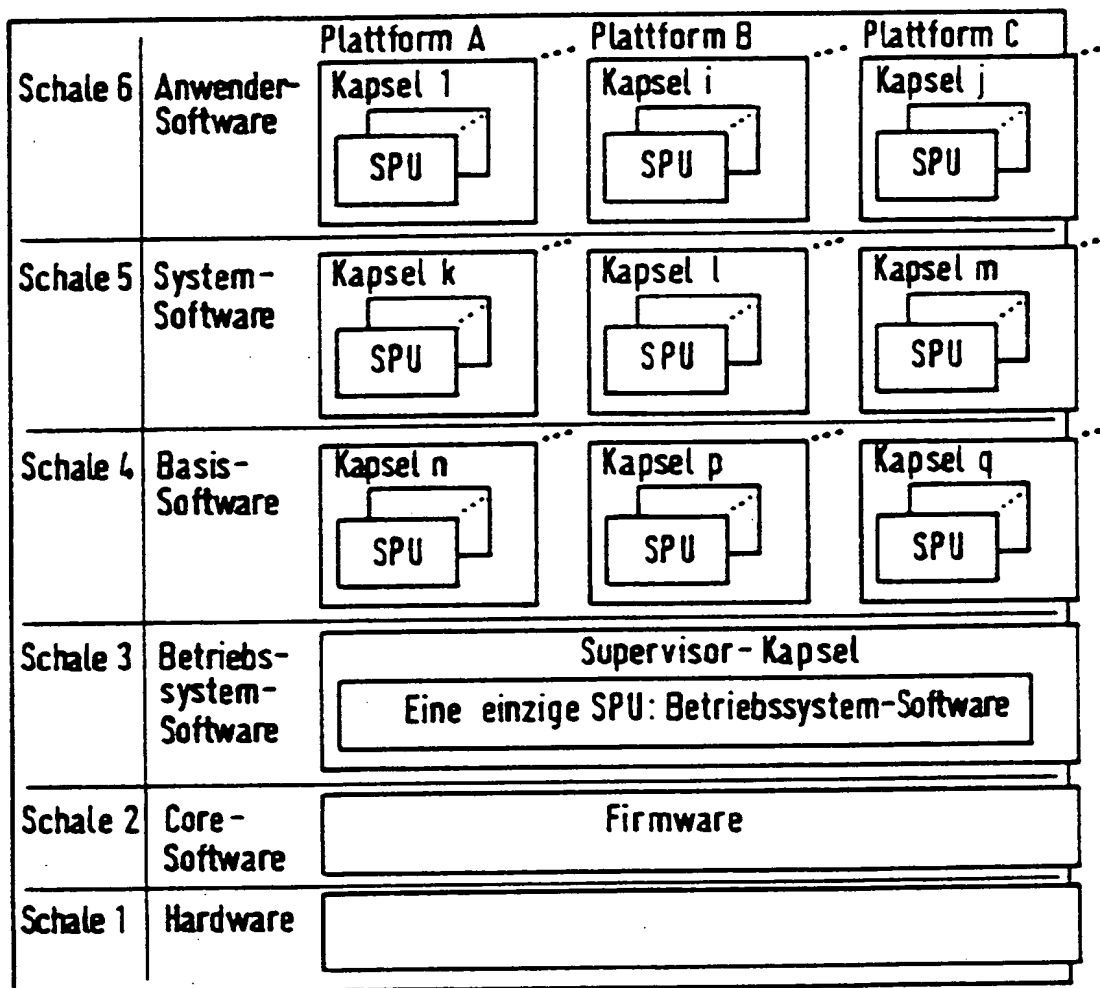
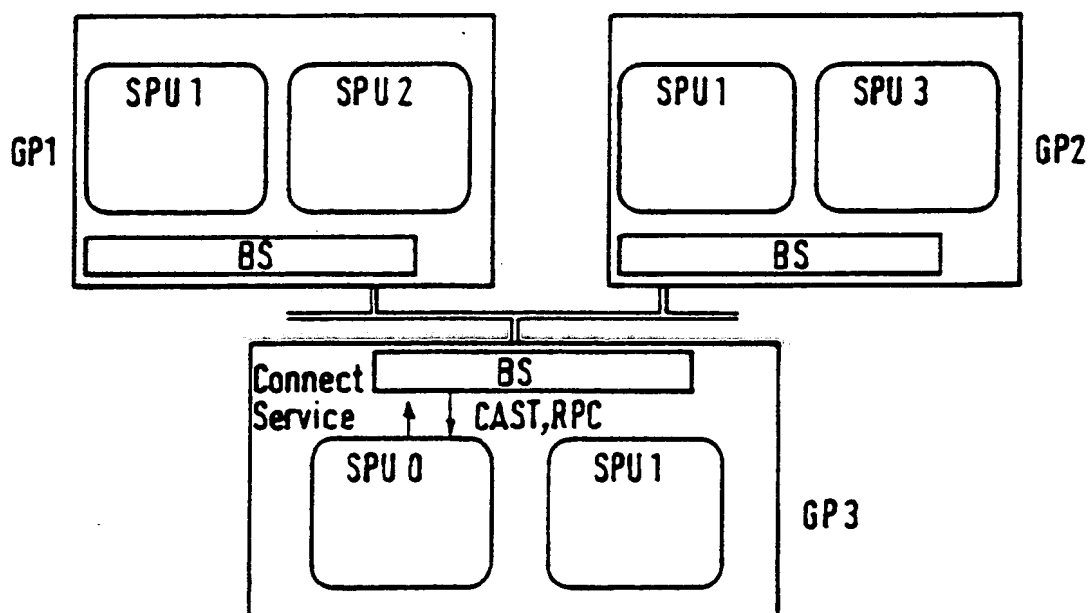


FIG 2



3 / 4

FIG 3

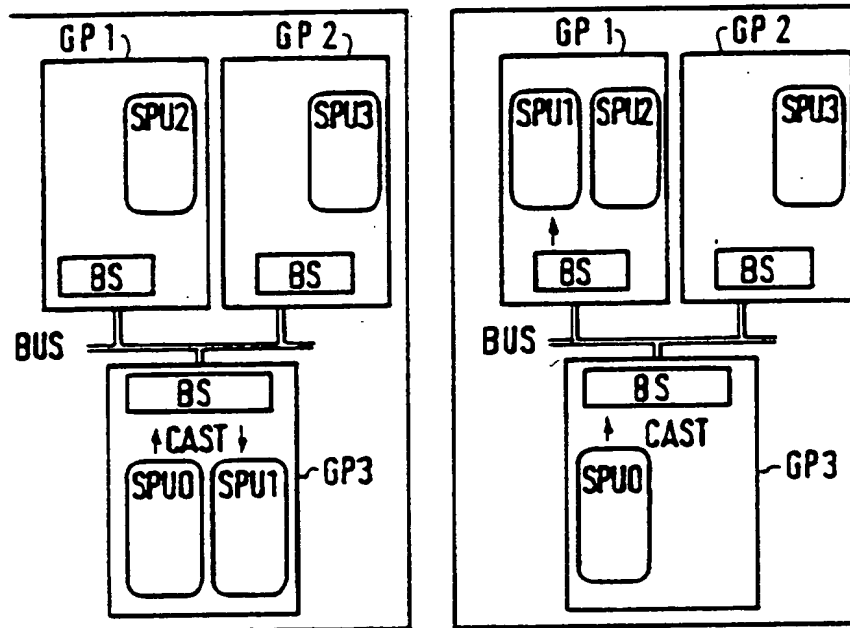


FIG 4

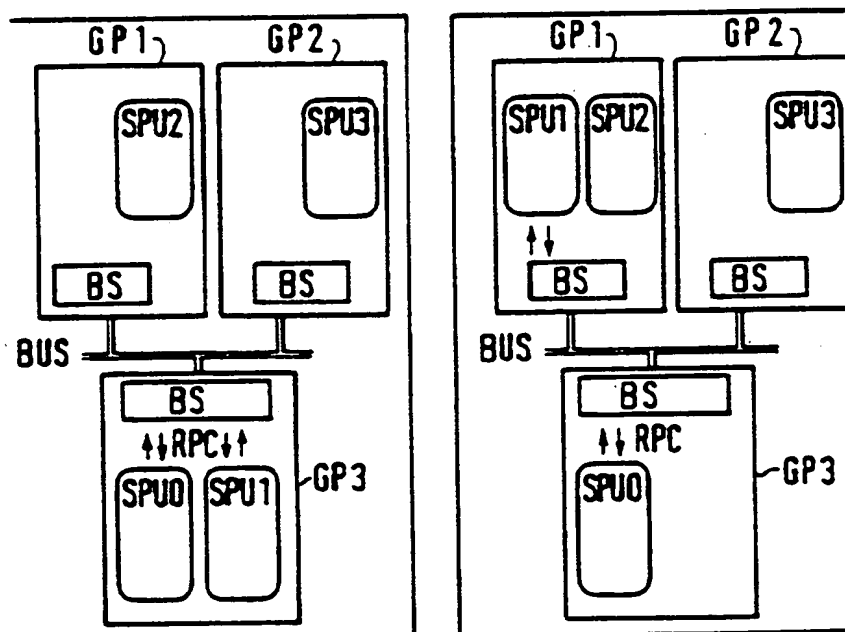
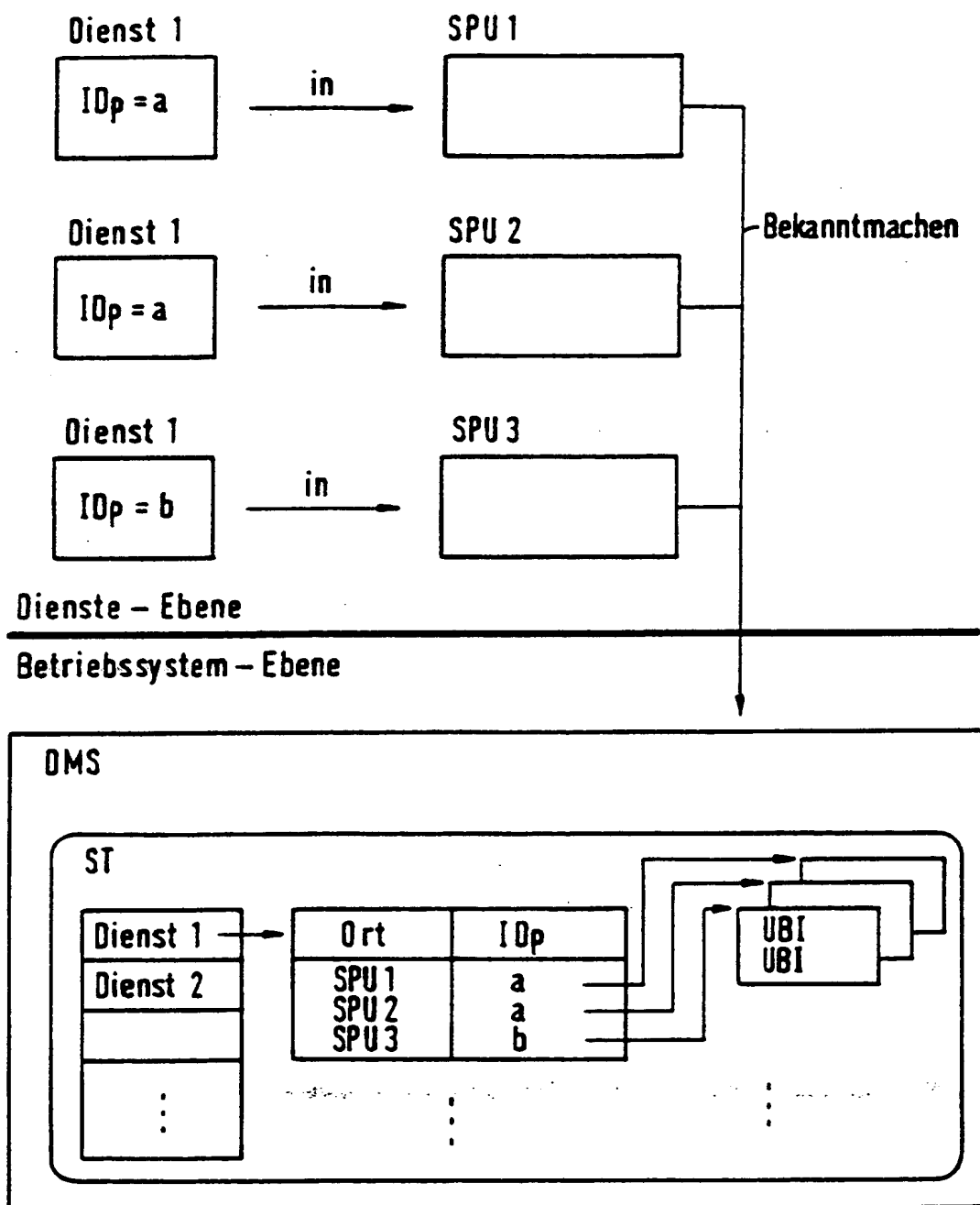


FIG 5



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)